#10

Claim Amendment Summary

Claims pending

- At time of the Action: 1-17, 19-35, 37-47, and 49-79.
- After this Response: 1-17, 19-35, 37-47, and 49-79.

Canceled claims: none.

Amended claims: none.

New claims: none.

Pending claims are listed as follows:

B1

1. **(PREVIOUSLY PRESENTED)** A method of formatting a message for exchange between entities on a network, the method comprising:

generating a message envelope;

generating contents of the message envelope, the contents comprising data structures, each data structure identifies which entity is intended to process the data structure when that entity receives the message envelope over the network.

2. **(PREVIOUSLY PRESENTED)** A method as recited in claim 1, wherein each data structure specifies whether the entity that is intended to process the data structure must understand such data structure.

3. **(ORIGINAL)** A method as recited in claim 1, wherein:

the message envelope has beginning and ending envelope tags;

the contents of the message envelope is between the envelope tags.

4. **(PREVIOUSLY PRESENTED)** A method as recited in claim 1, wherein the contents include:

a header data structure;

a body data structure, the body data structure including message data.

5. **(ORIGINAL)** A method as recited in claim 4, wherein:

the header data structure has beginning and ending header tags;

the body data structure has beginning and ending body tags.

6. **(PREVIOUSLY PRESENTED)** A method as recited in claim 4, wherein:

the header data structure is intended for at least one intermediate entity;

the body data structure is intended for a destination entity.

7. **(ORIGINAL)** A method as recited in claim 1 further comprising sending the message envelope to an entity on a network.

8. **(PREVIOUSLY PRESENTED)** A method as recited in claim 1, wherein at least one of the data structures includes a request for an entity to perform a task, wherein the data structures lack executable instructions for performing the task.

9. **(ORIGINAL)** A method as recited in claim 1, wherein the data structures are expressed in a markup language.

10. (ORIGINAL) A method as recited in claim 1, wherein the data structures are expressed in XML.

11. (ORIGINAL) A method as recited in claim 1 further comprising:

formatting the message envelope for sending over a network using HTTP;

sending the message envelope to an entity on the network by using HTTP.

12. (ORIGINAL) A method as recited in claim 1 further comprising:

binding the message envelope into a HTTP request;

sending the message envelope to an entity on the network by using HTTP.

13. (ORIGINAL) A method as recited in claim 1 further comprising:

binding the message envelope into a HTTP response;

sending the message envelope to an entity on the network by using HTTP.

14. (ORIGINAL) A method as recited in claim 3, wherein the envelope tags identify the message envelope.

15. (PREVIOUSLY PRESENTED) A method as recited in claim 5, wherein the header tags identify the header data structure.

16. (PREVIOUSLY PRESENTED) A method as recited in claim 5, wherein the body tags identify the body data structure.

17. (PREVIOUSLY PRESENTED) A method as recited in claim 4, wherein the message envelope has the following format:

<Envelope label>

<Header label>

header data

</Header label>

<Body label>

message data

</Body label>

</Envelope label>

the <Envelope label> being a beginning envelope tag, the </Envelope label> being an ending envelope tag, and the Envelope label identifying the message envelope;

the <Header label> being a beginning header tag, the </Header label> being an ending header tag, the Header label identifying the header data structure;

the <Body label> being a beginning body tag, the </Body label> being an ending body tag, and the Body label identifying the body data structure;

the header data being expressed in XML;

the message data being expressed in XML.

18. (CANCELLED).

19. (PREVIOUSLY PRESENTED) A method of delivering a message over a network, the method comprising:

transmitting a message envelope of a message from an origin entity to a destination entity via one or more intermediate entities on the network;

the message envelope having contents comprising data structures, each data structure identifies which entity is intended to process the data structure when that entity receives the message envelope over the network.

20. (PREVIOUSLY PRESENTED) A method as recited in claim 19, wherein each data structure specifies whether the entity that is intended to process the data structure must understand such data structure when that entity receives the message envelope over the network.

21. (ORIGINAL) A method as recited in claim 19, wherein:

the message envelope has beginning and ending envelope tags;

the contents of the message envelope is between the envelope tags.

22. (PREVIOUSLY PRESENTED) A method as recited in claim 19, wherein the contents include:

a header data structure;

a body data structure, the body data structure including message data.

23. (ORIGINAL) A method as recited in claim 22, wherein:

the header data structure has beginning and ending header tags;

the body data structure has beginning and ending body tags.

24. (ORIGINAL) A method as recited in claim 22, wherein:

the header data structure is intended for at least one intermediate entity;

the body data structure is intended for a destination entity.

25. (PREVIOUSLY PRESENTED) A method as recited in claim 19, wherein at least one of the data structures includes a request for an entity to perform a task, wherein the data structures lack executable instructions for performing the task.

26. (ORIGINAL) A method as recited in claim 19, wherein at least one of the data structures includes a request for an intermediate entity to perform a task.

27. (ORIGINAL) A method as recited in claim 19, wherein the data structures are expressed in a markup language.

28. (ORIGINAL) A method as recited in claim 19, wherein the data structures are expressed in XML.

29. (ORIGINAL) A method as recited in claim 19 further comprising:

formatting the message envelope for sending over a network using HTTP;

sending the message envelope to an entity on the network by using HTTP.

30. (ORIGINAL) A method as recited in claim 19 further comprising:

binding the message envelope into a HTTP request;

sending the message envelope to an entity on the network by using HTTP.

31. (ORIGINAL) A method as recited in claim 19 further comprising:

binding the message envelope into a HTTP response;

sending the message envelope to an entity on the network by using HTTP.

32. (ORIGINAL) A method as recited in claim 21, wherein the envelope tags identify the message envelope.

B1

33. (PREVIOUSLY PRESENTED) A method as recited in claim 23, wherein the header tags identify the header data structure.

34. (PREVIOUSLY PRESENTED) A method as recited in claim 23, wherein the body tags identify the body data structure.

35. (PREVIOUSLY PRESENTED) A method as recited in claim 22, wherein the message envelope has the following format:

<Envelope label>

<Header label>

header data

</Header label>

<Body label>

message data

</Body label>

</Envelope label>

the <Envelope label> being a beginning envelope tag, the </Envelope label> being an ending envelope tag, and the Envelope label identifying the message envelope;

the <Header label> being a beginning header tag, the </Header label> being an ending header tag, the Header label identifying the header data structure;

the <Body label> being a beginning body tag, the </Body label> being an ending body tag, and the Body label identifying the body data structure;

the header data being expressed in XML;

the message data being expressed in XML.

36. (CANCELLED)

37. (PREVIOUSLY PRESENTED) A method of parsing a message received by a receiving entity over a network from a sending entity, the method comprising:

parsing a message envelope;

parsing contents of the message envelope, the contents comprising data structures, each data structure identifies which entity is intended to process the data structure when that entity receives the message envelope over the network.

38. (PREVIOUSLY PRESENTED) A method as recited in claim 37, wherein each data structure specifies whether the entity that is intended to process the data structure must understand such data structure when that entity receives the message envelope over the network.

39. (PREVIOUSLY PRESENTED) A method as recited in claim 38 further comprising if the entity that is intended to process the data structure does not understand such data structure, sending a response message to the sending entity that indicates that the entity did not understand such data structure.

40. (ORIGINAL) A method as recited in claim 37 further comprising sending a response message to the sending entity on the network.

B1

41. (ORIGINAL) A method as recited in claim 37, wherein:

the message envelope has beginning and ending envelope tags;

the contents of the message envelope is between the envelope tags.

42. (PREVIOUSLY PRESENTED) A method as recited in claim 37, wherein the contents include:

a header data structure;

a body data structure, the body data structure including message data.

43. (ORIGINAL) A method as recited in claim 42, wherein:

the header data structure has beginning and ending header tags;

the body data structure has beginning and ending body tags.

44. (ORIGINAL) A method as recited in claim 42, wherein:

the header data structure is intended for at least one intermediate entity;

the body data structure is intended for a destination entity.

45. **(PREVIOUSLY PRESENTED)** A method as recited in claim 37, wherein at least one of the data structures includes a request for an entity to perform a task, wherein the data structures lack executable instructions for performing the task.

46. **(ORIGINAL)** A method as recited in claim 37, wherein the data structures are expressed in a markup language.

B1

47. **(ORIGINAL)** A method as recited in claim 37, wherein the data structures are expressed in XML.

48. **(CANCELLED)**

49. **(PREVIOUSLY PRESENTED)** A computer-readable storage medium having computer-executable instructions that, when executed by a computer, performs a method of formatting a message for exchange between entities on a network, the method comprising:

generating a message envelope;

generating contents of the message envelope, the contents comprising data structures, each data structure identifies which entity is intended to process the data structure when that entity receives the message envelope over the network.

50. **(PREVIOUSLY PRESENTED)** A computer-readable storage medium having computer-executable instructions that, when executed by a

computer, performs a method of delivering a message between entities on a network, the method comprising:

transmitting a message envelope of a message from an origin entity to a destination entity via one or more intermediate entities on the network;

the message envelope having contents comprising data structures, each data structure identifies which entity is intended to process the data structure when that entity receives the message envelope over the network.

B1

51. (PREVIOUSLY PRESENTED) A computer-readable storage medium having computer-executable instructions that, when executed by a computer, performs a method of parsing a message received by a receiving entity over a network from a sending entity, the method comprising:

parsing a message envelope of a message;

parsing contents of the message envelope, the contents comprising data structures, each data structure identifies which entity is intended to process the data structure when that entity receives the message envelope over the network.

52. (PREVIOUSLY PRESENTED) A message exchange apparatus comprising:

a processor;

a message formatter executable on the processor to:

generate a message envelope of a message;

generate contents of the message envelope, the contents comprising data structures, each data structure identifies which entity is intended to

process the data structure when that entity receives the message envelope over the network.

53. (PREVIOUSLY PRESENTED) A message exchange apparatus comprising:

a processor;

a message deliverer executable on the processor to:

transmit a message envelope of a message from an origin entity to a destination entity via one or more intermediate entities on the network;

the message envelope having contents comprising data structures, each data structure identifies which entity is intended to process the data structure when that entity receives the message envelope over the network.

54. (PREVIOUSLY PRESENTED) A message exchange apparatus comprising:

a processor;

a message parser executable on the processor to:

parse a message envelope of a message;

parse contents of the message envelope, the contents comprising data structures, each data structure identifies which entity is intended to process the data structure when that entity receives the message envelope over the network.

55. (PREVIOUSLY PRESENTED) A message envelope generated in accordance with the following acts:

providing a sending entity in communication with a network of entities;

generating contents of the message envelope of a message, the contents comprising:

a header data structure which identifies an intermediate entity as that which is intended to process the header data structure and whether that intermediate entity must understand such data structure; and

a body data structure which identifies a destination entity as that which is intended to process the body data structure.

B1

56. (ORIGINAL) A message envelope as recited in claim 55, wherein the data structures are expressed in a markup language.

57. (ORIGINAL) A message envelope as recited in claim 55, wherein the data structures are expressed in XML.

58. (PREVIOUSLY PRESENTED) A modulated data signal having computer-executable instructions embodied thereon comprising:

a header data structure which identifies an intermediate entity, over a network of entities, as that which is intended to process the header data structure and whether that intermediate entity must understand such data structure; and

a body data structure which identifies the destination entity as that which is intended to process the body data structure.

59. (ORIGINAL) A modulated data signal as recited in claim 58, wherein the data structures are expressed in a markup language.

60. (ORIGINAL) A modulated data signal as recited in claim 58, wherein the data structures are expressed in XML.

61. (PREVIOUSLY PRESENTED) A computer-readable medium having a data structure embodied thereon comprising:

a header data-structure section which identifies an intermediate entity, over a network of entities, as that which is intended to process the header data-structure section and whether that intermediate entity must understand such data-structure section; and

a body data-structure section which identifies the destination entity as that which is intended to process the body data-structure section.

62. (ORIGINAL) A computer-readable medium as recited in claim 61, wherein the data-structure sections are expressed in a markup language.

63. (ORIGINAL) A computer-readable medium as recited in claim 61, wherein the data-structure sections are expressed in XML.

64. (PREVIOUSLY PRESENTED) A method of formatting a message for exchange between entities on a network, the method comprising:

generating a message envelope of a message, the message comprising at least one request by one entity on a network of another entity on the network to perform a task;

generating contents of the message envelope, the contents comprising data structures, each data structure identifies which entity is intended to process the data structure when that entity receives the message envelope over the network.

65. (PREVIOUSLY PRESENTED) A method as recited in claim 64, wherein each data structure specifies whether the entity that is intended to process the data structure must understand such data structure.

66. (PREVIOUSLY PRESENTED) A method as recited in claim 64, wherein each data structure specifies whether the entity that is intended to process the data structure must respond if it does not understand such data structure.

67. (PREVIOUSLY PRESENTED) A method as recited in claim 64, wherein:

the message envelope has beginning and ending envelope tags;

the contents of the message envelope is between the envelope tags.

68. (PREVIOUSLY PRESENTED) A method as recited in claim 64, wherein the contents include:

a header data structure;

a body data structure, the body data structure including message data.

69. (PREVIOUSLY PRESENTED) A method as recited in claim 68, wherein:

the header data structure has beginning and ending header tags;

the body data structure has beginning and ending body tags.

70. (PREVIOUSLY PRESENTED) A method as recited in claim 68, wherein:

the header data structure is intended for at least one intermediate entity;

the body data structure is intended for a destination entity.

71. (PREVIOUSLY PRESENTED) A method as recited in claim 64 further comprising sending the message envelope to an entity on a network.

B1

72. (PREVIOUSLY PRESENTED) A method as recited in claim 64, wherein at least one of the data structures includes a request for an entity to perform a task, wherein the data structures lack executable instructions for performing the task.

73. (PREVIOUSLY PRESENTED) A method as recited in claim 64, wherein the data structures are expressed in a markup language.

74. (PREVIOUSLY PRESENTED) A method as recited in claim 64, wherein the data structures are expressed in XML.

75. (PREVIOUSLY PRESENTED) A method as recited in claim 64 further comprising:

formatting the message envelope for sending over a network using HTTP;

sending the message envelope to an entity on the network by using HTTP.

76. (PREVIOUSLY PRESENTED) A method as recited in claim 64 further comprising:

binding the message envelope into a HTTP request;

sending the message envelope to an entity on the network by using HTTP.

77. (PREVIOUSLY PRESENTED) A method as recited in claim 64 further comprising:

binding the message envelope into a HTTP response;

sending the message envelope to an entity on the network by using HTTP.

78. (PREVIOUSLY PRESENTED) A method as recited in claim 69, wherein the header tags identify the header data structure.

79. (PREVIOUSLY PRESENTED) A method as recited in claim 69, wherein the body tags identify the body data structure.